05 | 数据探索：怎样从数据中找到用户的RFM值？

2021-09-08 黄佳

《零基础实战机器学习》 课程介绍 >

讲述：黄佳

时长 16:22 大小 14.99M

你好，我是黄佳。

今天我们正式进入“业务场景闯关篇”模块。我在开篇词中介绍过，在这个模块中，我会围绕电商场景下的运营环节，带你挑战 5 个关卡：获客关、变现关、激活关、留存关和裂变关，帮你逐步掌握机器学习的相关知识和实操技巧。今天，我们就从第一关“获客关”开始！

人们常说移动互联网的运营已经进入了下半场，几乎所有的企业都希望能用更优质的产品和更精准的服务留住用户，这就需要制定出合适的获客策略。而要做到这一点，前提就是为用户精准画像，也就是根据用户的人口统计信息和消费行为数据，给用户分组，然后推测出用户的消费习惯和价值高低。

所以，为用户分组、画像，找到不同用户的特点，进而挖掘出哪些才是最有价值的用户，是目前互联网大厂中的数据分析师和机器学习工程师常做的工作。既然如此，那么我们就在获客关，结合一个具体的电商项目，来看看怎么根据用户的基本信息和消费行为数据，给用户分组画像。

定义问题

按照我们前两讲所说的机器学习“实战 5 步”，我们首先要做的就是，把项目的问题定义清楚。在我们这个项目中，你可以想象自己就职于一家名为“易速鲜花”的创业公司，担任这家公司的运营团队机器学习工程师。你现在要接手的第一个项目就是为公司的用户分组画像。

现在，你们的运营部门已经准备好了过往用户的基本信息和消费行为数据，你可以在这里下载获取它，然后用 Excel 打开，就能看到这个数据集中所包含的详细信息。

	A	B	C	D	E	F	G	H
1	订单号	产品码	消费日期	产品说明	数量	单价	用户码	城市
2	536374	21258	6/1/2020 9:09	五彩玫瑰五支装	32	10.95	15100	北京
3	536376	22114	6/1/2020 9:32	茉莉花白色25枝	48	3.45	15291	上海
4	536376	21733	6/1/2020 9:32	教师节向日葵3枝尤加利5枝	64	2.55	15291	上海
5	536378	22386	6/1/2020 9:37	百合粉色10花苞	10	1.95	14688	北京
6	536378	85099C	6/1/2020 9:37	橙黄香槟色康乃馨	10	1.95	14688	北京
7	536378	21033	6/1/2020 9:37	满天星干花花束	10	2.95	14688	北京

既然我们要从这份销售订单数据中，为用户分组画像，那么有一个关键的问题就是，我们以什么为依据，给用户做分组？

我们知道，用户的消费行为本身是不能直接用于数据分析的，这就需要我们把用户的行为转化成具体的数值，也就是可量化的指标，让我们对用户有更直观的认识。而且我们还可以将这些指标用在数据分析、广告精准投放、产品推荐系统等多个运营场景，来提升我们产品和服务的精准度。

这个具体的数值呢，就是 RFM（Recency、Frequency、Monetary），它是用户画像的衍生品，也是目前很多互联网厂商普遍采用的分析方式。具体来看：

R 是新进度，代表自用户上次消费以来的天数。这个指标其实也就是用户的热乎度，比如说你上次使用 App 是一个月前，你的系统里 R 值就是 30。如果你今天又用了一次 App，那 R 值就会立刻被更新为 1；

F 是消费频率，代表用户是否频繁使用服务。这也是用户黏性的风向标；

M 是消费金额，代表用户在一段时间内消费的总金额，这个不难理解。

通过用户消费行为的新进度、消费的总体频率，还有消费总金额这三项指标，我们可以将用户划分为不同的类别或集群，来直观地描述用户的价值。什么意思呢？简单来说就是，只要我们从用户的基本信息和消费行为数据中得出 RFM 值，就可以根据它对用户分组画像了。

所以，我们这个项目整体上可以分为两个阶段：第一个阶段是求出 RFM 值，第二个阶段就是利用 RFM 值，给用户分组画像，进而绘制出高价值、中等价值和低价值用户的分布情况。我们今天这节课的目标就是解决第一阶段的问题。

搞清楚问题后，现在我们来看看这份数据的整体状况，根据需要对数据进行一个初步的预处理。

数据预处理

由于已经完成了数据收集工作，我们现在直接用 Pandas 中的 read_csv 工具，把这个原始数据集读入到 Pandas 的 DataFrame 中，用 Dataframe 形式呈现出来。关于 Pandas 包及其导入方式，如果你感觉比较陌生，可以再回顾一下第三讲的内容。

复制代码

```
import pandas as pd #导入Pandas
df_sales = pd.read_csv('易速鲜花订单记录.csv') #载入数据
df_sales.head() #显示头几行数据
```

运行这段代码后，输出如下：

	订单号	产品码	消费日期	产品说明	数量	单价	用户码	城市
0	536374	21258	6/1/2020 9:09	五彩玫瑰五支装	32	10.95	15100	北京
1	536376	22114	6/1/2020 9:32	茉莉花白色25枝	48	3.45	15291	上海
2	536376	21733	6/1/2020 9:32	教师节向日葵3枝尤加利5枝	64	2.55	15291	上海
3	536378	22386	6/1/2020 9:37	百合粉色10花苞	10	1.95	14688	北京
4	536378	85099C	6/1/2020 9:37	橙黄香槟色康乃馨	10	1.95	14688	北京

从图中，你可以看到这个数据集主要包含了订单号、产品码、消费日期，产品说明、数量、单价、用户码和城市等字段。因为我们求的是用户消费行为的新进度（R）、消费的总体频率（F），还有消费总金额（M），所以在这些信息中，我们重点关注的是以下几个数据：

用户码

单价

（订单中产品的）数量

消费日期

为了对公司运营团队交给我们的数据集有一个宏观上的了解，下面我们先做一个整体的数据可视化，看看这个数据集所覆盖的消费日期跨度是怎样的。

1. 数据可视化

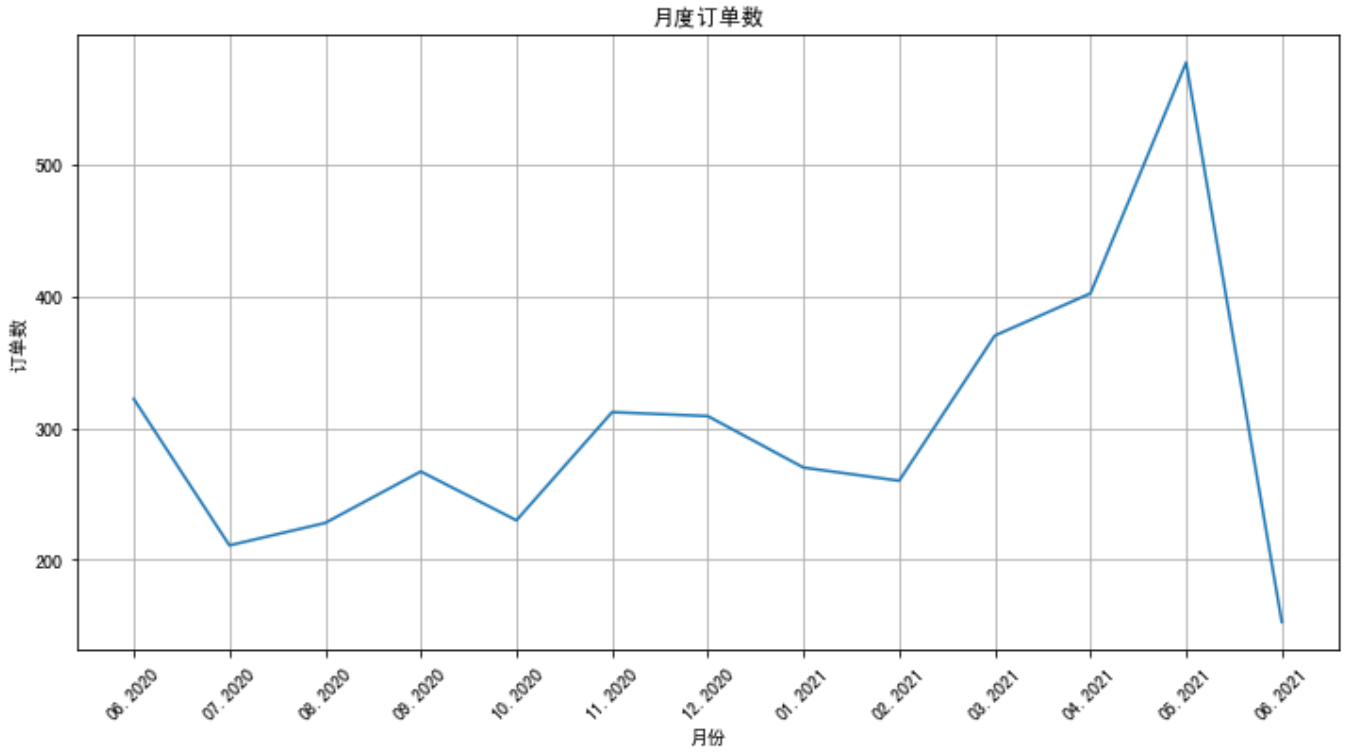
我们直接来看具体的实现代码：

复制代码

```
import matplotlib.pyplot as plt #导入Matplotlib的pyplot模块
#构建月度的订单数的DataFrame
df_sales['消费日期'] = pd.to_datetime(df_sales['消费日期']) #转化日期格式
df_orders_monthly = df_sales.set_index('消费日期')['订单号'].resample('M').nuniq
#设定绘图的画布
ax = pd.DataFrame(df_orders_monthly.values).plot(grid=True,figsize=(12,6),lege
ax.set_xlabel('月份') # X轴label
ax.set_ylabel('订单数') # Y轴Label
ax.set_title('月度订单数') # 图题
#设定X轴月份显示格式
plt.xticks(
    range(len(df_orders_monthly.index)),
    [x.strftime('%m.%Y') for x in df_orders_monthly.index],
    rotation=45)
plt.show() # 绘图
```

在这段代码中，你需要注意的是，我们这里使用了 Pandas 的 to_datetime 这个 API，把原始消费日期转换成了能处理的格式。而在 df_orders_monthly 中，则是求出了每一个月的订单数量。再往后的代码是具体的绘图工具使用，我就不过多说明了。

输出如下：



我们看到，这个数据集收集了“易速鲜花”公司一整年的订单量。所以啊，我们要求的消费额 M，实际上是每个用户一整年的总消费额。

你可能已经注意到，在最后一个月，也就是 2021 年 6 月，订单量突然大幅下降。其实这是因为运营人员拉这个表的时候，正是 6 月的第一个礼拜。所以，6 月的数据虽然不全，但并不会影响我们对用户 RFM 值的分析。

2. 数据清洗

完成了初步可视化之后，我们来清洗一下数据。请你注意，在刚才的可视化过程中，我们已经完成了对消费日期的观察，并没有发现什么异常。所以，现在我们重点要处理的是用户码、单价和（订单中产品的）数量。

首先，我们用 Pandas 中的 drop_duplicates 方法把完全相同的重复数据行删除掉。

复制代码

```
df_sales = df_sales.drop_duplicates() #删除重复的数据行
```

我之前介绍过，你还可以用 DataFrame 的 isna().sum() 函数，来统计 NaN 的个数。当然，如果你尝试了这个方法，会发现这个数据集中没有 NaN 值。

此外，对于数量、金额等类型的数据，我们还常常会使用 describe 方法来查看这些字段的统计信息是否有脏数据。

```
plain
df_sales.describe() #df_sales的统计信息
```

复制代码

输出如下：

	数量	单价	用户码
count	87180.000000	87180.000000	87180.000000
mean	10.006435	3.593587	15338.503774
std	48.769498	133.444401	392.001142
min	-9360.000000	0.000000	14681.000000
25%	2.000000	1.250000	15022.000000
50%	4.000000	1.950000	15335.000000
75%	12.000000	3.750000	15674.000000
max	3114.000000	38970.000000	16019.000000

在图中你可以看到这个数据集中，共有 9 万多行的数据（count 统计数据条目的数量），每条数据的平均采购数量是 10（mean 统计均值），商品平均单价是 3.575 元左右。

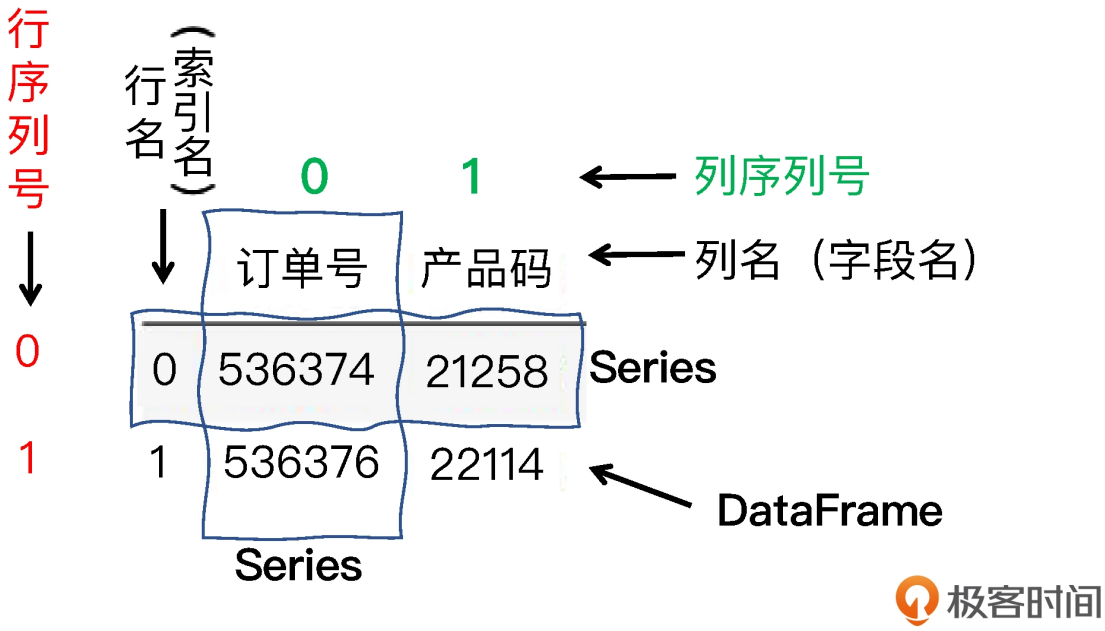
在概览中我们发现，（订单中产品的）数量的最小值（min）是一个负数（-9360），这显然是不符合逻辑的，所以我们要把这种脏数据清洗掉。具体的处理方式是，用 loc 属性通过字段名（也就是列名）访问数据集，同时只保留“数量”字段大于 0 的数据行：

```
df_sales = df_sales.loc[df_sales['数量'] > 0] #清洗掉数量小于等于0的数据
```

复制代码

在 DataFrame 对象中，loc 属性是通过行、列的名称来访问数据的，我们做数据预处理时会经常用到；还有一个常被用到的属性是 iloc，它是通过行列的位置（也就是序号）来访问数据的。

由于数据行的名称往往也是数值，这两者非常容易被弄混。所以，在这里我特地给你做了一张图，来说明什么是行列名、什么是行列序号，帮你弄清楚它们的区别。



这个语句执行后，如果你再用一次 describe 方法，就会看到新的最小数量为 1，这就对了。

	数量	单价	用户码
count	85354.000000	85354.000000	85354.000000
mean	10.641833	3.007199	15338.156712
std	33.788964	15.330760	392.798853
min	1.000000	0.000000	14681.000000
25%	2.000000	1.250000	15021.000000
50%	4.000000	1.950000	15334.000000
75%	12.000000	3.750000	15674.000000
max	3114.000000	3155.950000	16019.000000

到这里，我们就完成了数据清洗工作，这份数据集中已经没有可以剔除的脏数据了。接下来，我们就看看怎么从中求得 RFM 值。

求 RFM 值

不过在计算 RFM 之前，我想请你再回顾一下我们整个项目的目标。我们的目标就是建立一个机器学习模型，给用户做价值分组。而其中，RFM 实际上就是我们构建模型所需的关键

特征数据。

讲到这里，我想你应该很清楚了，虽说我们这节课要求出 RFM 的值，但实际上，我们是在完成机器学习项目中的“特征工程”环节，也就是对原始数据集中的信息进行选择、提取、合并、加工、转换，甚至是基于原始信息构建出新的、对于模型的训练更具有意义的特征。

明白了这一点，我们就可以继续往下走了。那怎么计算这个 RFM 值呢？

其实，这里的 R 值（最近一次消费的天数）和 F 值（消费频率），我们通过数据集中的消费日期就能得到，但是对于 M 值（消费金额），你会发现数据集中有采购数量，有单价，但是还没有每一笔消费的总价。

因此，我们通过一个语句对原有的数据集进行一个小小的扩展。在 df_sales 这个 DataFrame 对象中增加一个数据列计算总价，总价等于由单价乘以数量：

复制代码

```
df_sales['总价'] = df_sales['数量'] * df_sales['单价'] #计算每单的总价
df_sales.head() #显示头几行数据
```

输出如下：

	订单号	产品码	消费日期	产品说明	数量	单价	用户码	城市	总价
0	536374	21258	6/1/2020 9:09	五彩玫瑰五支装	32	10.95	15100	北京	350.4
1	536376	22114	6/1/2020 9:32	茉莉花白色25枝	48	3.45	15291	上海	165.6
2	536376	21733	6/1/2020 9:32	教师节向日葵3枝尤加利5枝	64	2.55	15291	上海	163.2
3	536378	22386	6/1/2020 9:37	百合粉色10花苞	10	1.95	14688	北京	19.5
4	536378	85099C	6/1/2020 9:37	橙黄香槟色康乃馨	10	1.95	14688	北京	19.5

现在，在这个数据集中，**用户码**、**总价**和**消费日期**这三个字段，给我们带来了每一个用户的 R、F、M 信息。其中：

一个用户上一次购物的日期，也就是最新的消费日期，就可以转化成这个用户的 R 值；

一个用户下的所有订单次数之和，就是消费频率值，也就是该用户的 F 值；

把一个用户所有订单的总价加起来，就是消费金额值，也就是该用户的 M 值。

不过，我们目前的这个数据集是一个订单的历史记录，并不是以用户码为主键的数据表。而 R、F、M 信息是和用户相关的，每一个用户都拥有一个独立的 R 值、F 值和 M 值，所以，在计算 RFM 值之前，我们需要先构建一个用户层级表。

1. 构建用户层级表

构建用户层级表，简单来说就是生成一个以用户码为关键字段的 Dataframe 对象 df_user，然后在这个 Dataframe 对象中，逐步加入每一个用户的新近度（R）、消费频率（F）、消费金额（M），以及最终总的分组信息。

在代码实现上，我们用 Dataframe 的 unique() 这个 API，就能创建出以用户码为关键字段的用户层级表 df_user，然后我们再设定字段名，并根据用户码进行排序，最后显示出这个表，就可以了：

复制代码

```
df_user = pd.DataFrame(df_sales['用户码'].unique()) #生成以用户码为主键的结构df_use
df_user.columns = ['用户码'] #设定字段名
df_user = df_user.sort_values(by='用户码',ascending=True).reset_index(drop=True
df_user #显示df_user
```

请你注意，在上述语句中，reset_index(drop=True) 的意思是重置索引，生成新的默认数值类型索引，并且不保留原来的索引。运行这段代码后，我们得到这样的结果：

	用户码
0	14681
1	14682
2	14684
3	14687
4	14688
...	...
975	16015
976	16016
977	16017
978	16018
979	16019

980 rows × 1 columns

可以看到，我们一共有 980 个用户的数据。有了用户层级表，现在我们依次求出 RFM，让这个用户层级表的结构越来越完整。

2. 求出 R 值

我们知道，R 值代表自用户上次消费以来的天数，它与最近一次消费的日期相关。所以，用表中最新订单的日期（拉出来这张表的日期）减去上一次消费的日期，就可以确定对应用户的 R 值。

下面是具体的代码实现，在每行代码中都给出了详细的注释，而且 Pandas 的语句也都不难理解，你可以看一看：

复制代码

```
df_sales['消费日期'] = pd.to_datetime(df_sales['消费日期']) #转化日期格式
df_recent_buy = df_sales.groupby('用户码').消费日期.max().reset_index() #构建消费日
df_recent_buy.columns = ['用户码','最近日期'] #设定字段名
df_recent_buy['R值'] = (df_recent_buy['最近日期'].max() - df_recent_buy['最近日期
df_user = pd.merge(df_user, df_recent_buy[['用户码','R值']], on='用户码') #把上次
df_user.head() #显示df_user头几行数据
```

输出如下：

	用户码	R值
0	14681	70
1	14682	187
2	14684	25
3	14687	106
4	14688	7

R 值越大，说明该用户最近一次购物日距离当前日期越久，那么这样的用户就越是处于休眠状态。从表中可以看出来，编号为 14682 的用户已经有 187 天没有购物了。所以我们就可以判断这个用户呈现休眠态，很可能已经被别的购物平台所吸引了，也就是流失了。

3. 求出 F 值

类似地，我们还可以求出 F 值，并把 F 值添加至用户层级表：

```
df_frequency = df_sales.groupby('用户码').消费日期.count().reset_index() #计算每个
df_frequency.columns = ['用户码','F值'] #设定字段名称
df_user = pd.merge(df_user, df_frequency, on='用户码') #把消费频率整合至df_user结构
df_user.head() #显示头几行数据
```

这段代码的核心就是通过给消费日期做 count() 计数，来求出每一个用户的消费次数。最后，我们得到的输出结果如下：

	用户码	R值	F值
0	14681	70	7
1	14682	187	2
2	14684	25	390
3	14687	106	15
4	14688	7	324

4. **求出 M 值**

M 值很容易求出，它就是用户消费的总和：

```
df_revenue = df_sales.groupby('用户码').总价.sum().reset_index() #根据消费总额，构
df_revenue.columns = ['用户码','M值'] #设定字段名称
df_user = pd.merge(df_user, df_revenue, on='用户码') #把消费金额整合至df_user结构
df_user.head() #显示头几行数据
```

这段代码的核心是通过给用户每张订单的总价字段做 sum() 计数，来求出每一个用户的消费总和。最后，输出如下：

	用户码	R值	F值	M值
0	14681	70	7	498.95
1	14682	187	2	52.00
2	14684	25	390	1201.51
3	14687	106	15	628.38
4	14688	7	324	12130.60

到这里，我们就求出了每一个用户的 R、F、M 值。不过，这只是完成了特征工程环节，在下节课，我们要根据这三个维度的值给用户分组，这就需要聚类算法大显身手了。

总结一下

现在，我们来回顾一下这节课的重点。

我们这个项目的目标是为用户分组画像。要做分组画像，RFM 分析是一个不错的方法，它也是目前很多互联网厂商普遍采用的分析方式。

我们今天的重点是，根据用户码、总价和消费日期这三个字段，从消费历史数据中求出每位用户的 R、F、M 的值，这就好像给用户贴上了一堆数字化的标签。同时，**用户行为的量化，也可以视作是进一步做机器学习项目之前的特征工程**。这也为我们后续给用户做聚类分组做好了准备。

理解 RFM 分析的精髓，并利用好 RFM 三个指标，是我们这个项目实战的核心价值所在。求出了 RFM 值，就恭喜你闯过了我们获客阶段的数据探索这一关。

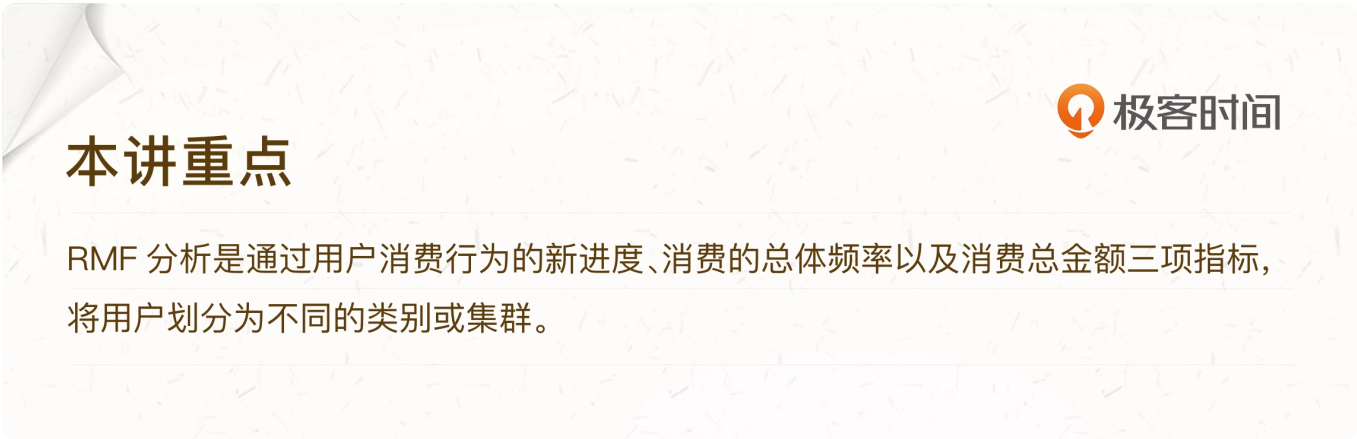
虽然我把实战的具体过程讲得很细，但我更希望你理解的是**“数据处理之道”**，而不是只学几个 Python 语句。Python 语句都很简单，但是怎么灵活应用数据，是我们一开始就应该领悟的“道”，因为只有对思路的深入理解，才能让我们走得更高、更远。

思考题

好，这节课就到这里了，最后，我给你留个思考题：

RFM 模型是衡量客户价值和客户创利能力的重要工具和手段，而且也被广泛应用在企业的获客、促销、客户关系管理等诸多营销、推广环节。请你谈一谈你对 RFM 的理解，以及这些指标可以应用于哪些业务场景。

欢迎你在留言区和我分享你的观点，如果你认为这节课的内容有收获，也欢迎把它分享给你的朋友，我们下一讲再见！



分享给需要的人，Ta订阅后你可得 20 元现金奖励

赞 7 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 04 | 实战5步（下）：怎么建立估计10万+软文点击率的模型？

下一篇 06 | 聚类分析：如何用RFM给电商用户做价值分组画像？

精选留言 (8)

写留言

AIK
2021-09-08

上班偷偷摸鱼，把老师分享的课程由粗到细品味了三遍，老师不仅技术高超，运营也这么牛，真是望尘莫及，在没订阅课程之前，我一点运营的知识都没有，看完今天的课程感觉自己又可以了。聊下老师的思考题，我想起当年的瑞幸和滴滴就这么搞的，通过下载APP注册，发送大量的消费券，获取很多的用户。然后经过时间的沉淀，利用RFM技术对用户进行分组，对于用户粘性较高的，进行精细化推荐。对于用户粘性较低的，发送消费券...
展开

作者回复: 十分感谢你的分享和肯定。也非常非常欣慰你能够在学习中得到收获，感受到进步，同时得到乐趣。我觉得这就是学习的上佳状态。学而时思之，不亦乐乎。

也谢谢你分享展示中文字体的经验！有需要的同学可以参考一下 😄

5

海林Lin
2021-09-09

课程很有收获，有个问题请教老师，为什么说RFM分析能够应用在获客环节呢，个人理解这个时候往往没有用户的行为数据
展开

作者回复: 嗯，挺好的问题。其实，从AARRR海盗模型的角度来说，获客、激活、留存、变现、增长循环，这5个环节是一个有机的整体，其实并没有一个明显的分割线。很多工具都可以重复使用在各个环节中。现在有一种说法，就是从AARRR到RARRA的模型变化，本质是从野蛮获取用户到精细化运营用户的视野转变。那么，因为现在大多数App获客阶段已经结束了，已经拥有了海量用户。那么，为这些用户画像，精细化运营就更重要。

回到你的问题：为什么RFM分析能够应用在获客环节？从机器学习的角度来讲，如果我能够利用一批老客户的特征（不一定是消费行为特征，也许是人口统计学特征），确定各个客户的价值。那么，我就可以用这个模型做指导，判断出新的具有类似特点的新客户价值有多大。对吧。所以这个工具可以应用在获客环节。

1 2

茜茜
2021-09-08

我觉得RFM适用于高频交易的场景，如零售，不适用于低频交易场景，如赛事演出票务。通过RFM将可以用户分为高中低等价值用户，在需要对某些产品进行营销推广时，可以将高价值用户定为主要营销群体，从而获得更高的订单转化率。但在计算RFM时，可能会面临以下问题：1.用户id与用户并不是一一对应或用户id不统一：如用户有多个账号，或者是多人使用该用户的id发生购买行为，或者部分消费记录未记入该用户，通过的RFM值...

展开

作者回复: 对的，具体问题具体分析，活用数据。

2

Jove
2021-09-14

RFM 模型感觉也可以运用到更多的场景，假设我要针对的是一群没有开通会员的用户，来提高付费率。
那么这里的 R 记忆强度；F 忠诚度；M 在这里就没有金额，或者说可以用用户的地理位置和手机型号等来代替。
按重要程度来看 F > R > M，也就是不同场景下，各自的权重是不一样的。

展开

作者回复: 是这样的，可灵活运用。

timothywei
2021-09-09

催更了，期待继续更新

展开

作者回复: 好，新篇明早准时上线。

吴悦
2021-09-08

正好 最近在做这个RFM模型 ，期待后面的聚类模式是怎么处理的

作者回复: 哈哈，好的，敬请期待。周五见。如果我的模型有值得商榷之处，也请分享您的经验和见解。大家可以一起学习讨论。

在路上
2021-09-08

佳哥好，看到github上更新了这节课的最新代码，非常开心，这对于我这种初次接触Jupyter和Python的学员来说太重要了，调试非常方便。
RFM 模型是我今天学到的新知识，F代表行为的频率，M代表行为的程度，如果行为是消费，M就是消费金额，如果行为是充值，M就是充值金额，F和M是行为的两个特征，就像我们用振频和振幅来描述振动的特征。R代表行为最近发生的时间，RFM从不同维度描...
展开

作者回复: 深为你认真学习的精神而感动！而且，你在学习实战的同时都有很多思考，难能可贵。

我们的数据集，和RFM这些维度，只是用抽象的、大家可以理解的数据启发一个思考，当你吧数据分析、可视化、机器学习的思路应用到公司的具体业务数据中，你能够发现数据能带来很多原来看不到的新启发和新灵感。这些东西只有我们在工作、学习的个人实践中去发现。

也欢迎大家多多分享自己的实践心得。

！null
2021-09-08

对于实战课，手不能懒，得敲一下熟练熟练。
展开

作者回复: 对，从我自己学习的过程中我也有这个体会。只有动手，才能够发现更多的奥秘。